

WHAT IS CLAIMED IS:

1. A method comprising:

generating a source document, the source document including at least one event;

associating meta information with one or more of the events;

transforming the events into one or more markup language specific representations of
5 the events, the transformation of an event being controlled at least in part by the associated
meta-information;

sending at least one markup language specific representation of the events to a
browser running on a client device; and

receiving from the client device one or more markup language specific events coded
10 as HTTP-request parameters.

2. The method of claim 1 wherein generating the source document comprises
generating the source document to include at least one generic, markup language
independent, event.

3. The method of claim 1 wherein the source document is a web document.

4. The method of claim 3 wherein the generic, markup language independent, event
is described in a generic, device-independent document description language based on XML.

5. The method of claim 4 wherein associating meta information comprises manually
associating meta information with one or more of the events.

6. The method of claim 1 wherein the meta information indicates alternative
25 representations of semantically one element.

7. The method of claim 1 wherein the meta information enables elements to be
declared to be optional and to be omitted on a client device with insufficient resources.

8. The method of claim 1 wherein transforming the events comprises automatically transforming the events.

9. The method of claim 1 further comprising fragmenting the source document into two or more subdocuments and transforming the fragments into one or more markup language specific representations appropriate to available resources of the client device and an execution environment of the client device.

10. The method of claim 1 wherein the one or more markup language specific representations comprise one or more of an HTML representation, a WML representation, and a cHTML representation.

11. The method of claim 1 wherein the generic events comprise one or more of a navigation event, an input event, a relation event, and a submission event.

12. A method comprising:
 generating a source document, the source document including at least one generic, markup language independent, event;
 manually associating meta information with one or more of the generic events;
 automatically transforming the source document generic events into one or more markup language specific representations of the source document events, the transformation of an event being controlled at least in part by the associated meta-information;
 sending at least one markup language specific representation of the events to a browser running on a client device; and
 receiving from the client device one or more markup language specific events coded as HTTP-request parameters.

13. An apparatus comprising a server device configured to:
 generate a source document, the source document including at least one generic, markup language independent, event;
 associate meta information with one or more of the events;

transform the events into one or more markup language specific representations of the events, the transformation of an event being controlled at least in part by the associated meta-information;

send at least one markup language specific representation of the events to a browser running on a client device; and

receive from the client device one or more markup language specific events coded as HTTP-request parameters.

14. The apparatus of claim 13 further comprising a server device configured to fragment the source document into two or more subdocuments and transform the fragments into one or more markup language specific representations appropriate to available resources of the client device and an execution environment of the client device.

15. An apparatus comprising:

an adaptation framework comprising:

an event dispatcher configured to process an incoming event and control an invocation of one or more processes based upon the event;

a fragment getter invoked by the event dispatcher and configured to retrieve a portion of a document from a local data store;

a processor invoked by the event dispatcher and configured to communicate with the fragment getter and configured to transform the document into a device specific format; and

a fragmentation filter invoked by the event dispatcher configured to fragment the document into one or more parts for display by a client device based upon an availability of one or more resources at the client device.

16. The apparatus of claim 15 wherein the adaptation framework further comprises a client recognizer configured to receive information from a client device and to receive device profile information.

17. The apparatus of claim 15 wherein the fragment getter is further configured to generate the document from a local data store based upon user profile data stored in a user profile.

5 18. The apparatus of claim 15 wherein the adaptation framework further comprises an image filter configured to adapt an image according to device profile information and user profile information.

10 19. The apparatus of claim 15 wherein the fragmentation filter further comprises a first fragmentation filter configured to manage caching of one or more fragments of the document and configured to perform a fragmentation of the document; and a fragmentation validation filter communicating with the first fragmentation filter configured to determine whether the fragments may be rendered on the client device without exceeding the resources of the client device, and if not, to enable further fragmentation by the first fragmentation
15 filter.

20 20. The apparatus of claim 15 wherein the one or more filters are executed based upon filter configuration data stored in a filter configuration file.

20 21. The apparatus of claim 15 wherein a next filter to be executed is determined by a current filter being executed.